

Static routing IP v4

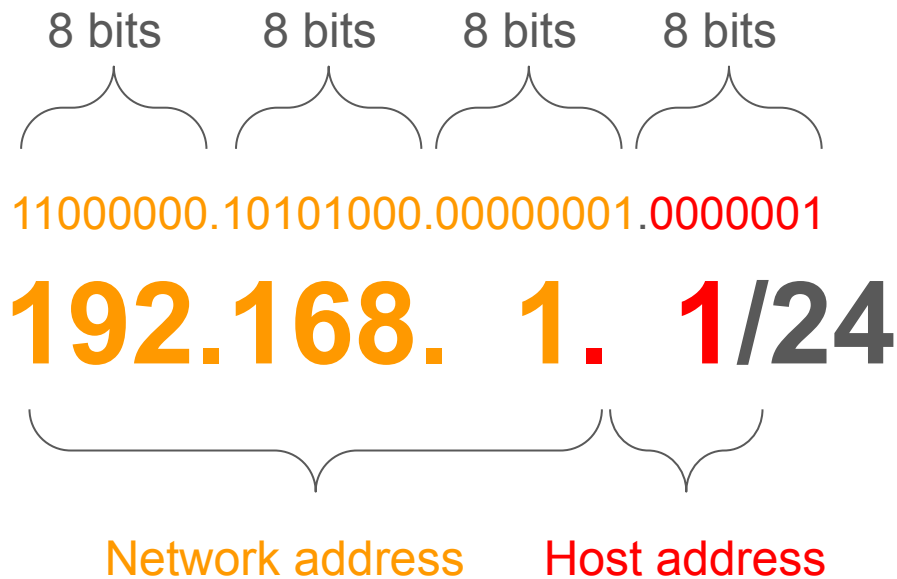
IPv4 address structure

- 32-bit long
- Network bits
- Host bits

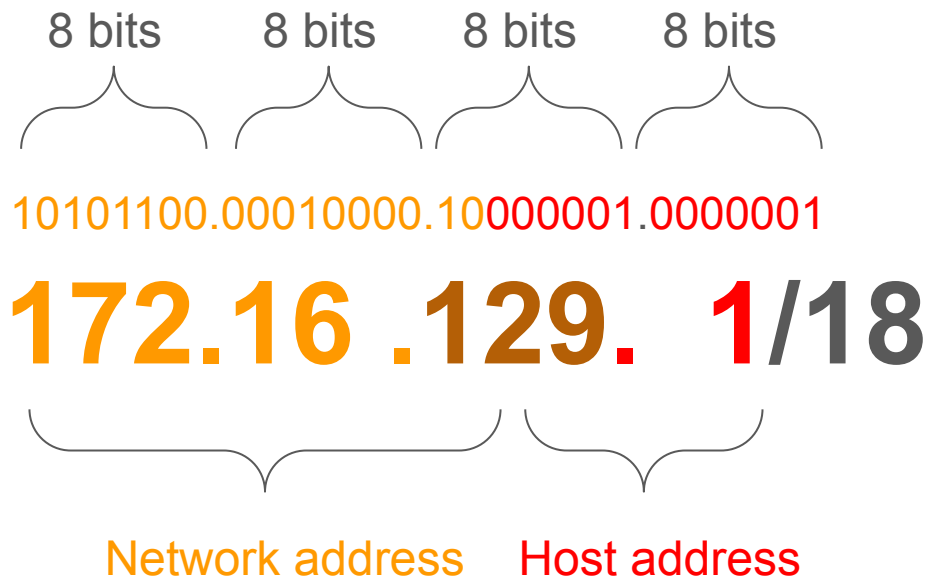
IPv4 address structure

- 32-bit long
- Network bits
- Host bits

192.168.1.1/24



- **192.168.1.0** - network address
- **192.168.1.1-254** - hosts in the subnet
- **192.168.1.255** - broadcast address



- 172.16.128.0 - network address
- 172.16.128.1- 172.16.191.254 - hosts in the subnet
- 172.16.191.255 - broadcast address

Unicast routing - when does it take place?

- Before packet is sent out, device calculates whether the target host belongs to a different network.
 - If it's in the same network, ARP packet is sent out to resolve IPv4 address into MAC address for the *direct* reachability
 - If it's in the different network, packet is sent out to the device acting as a gateway for that host

ARP - Address Resolution Protocol

- Every frame in Ethernet networks needs to have destination host MAC address. ARP is used to create IP-to-MAC mapping

No.	Time	Source	Destination	Protocol	Length	Info
3	8.269567544	50:00:00:01:00:01	50:00:00:07:00:04	ARP	60	Who has 192.168.29.1? Tell 192.168.29.2
4	8.270379907	50:00:00:07:00:04	50:00:00:01:00:01	ARP	60	192.168.29.1 is at 50:00:00:07:00:04
6	15.291529878	192.168.29.2	192.168.29.1	ICMP	114	Echo (ping) request id=0x0003, seq=0/0, ttl=255 (reply in 7)
7	15.291973553	192.168.29.1	192.168.29.2	ICMP	114	Echo (ping) reply id=0x0003, seq=0/0, ttl=255 (request in 6)
8	15.292843947	192.168.29.2	192.168.29.1	ICMP	114	Echo (ping) request id=0x0003, seq=1/256, ttl=255 (reply in 9)
9	15.293084869	192.168.29.1	192.168.29.2	ICMP	114	Echo (ping) reply id=0x0003, seq=1/256, ttl=255 (request in 8)
10	15.293813898	192.168.29.2	192.168.29.1	ICMP	114	Echo (ping) request id=0x0003, seq=2/512, ttl=255 (reply in 11)
11	15.294029527	192.168.29.1	192.168.29.2	ICMP	114	Echo (ping) reply id=0x0003, seq=2/512, ttl=255 (request in 10)
12	15.294758912	192.168.29.2	192.168.29.1	ICMP	114	Echo (ping) request id=0x0003, seq=3/768, ttl=255 (reply in 13)
13	15.295039421	192.168.29.1	192.168.29.2	ICMP	114	Echo (ping) reply id=0x0003, seq=3/768, ttl=255 (request in 12)
14	15.295792605	192.168.29.2	192.168.29.1	ICMP	114	Echo (ping) request id=0x0003, seq=4/1024, ttl=255 (reply in 15)
15	15.295933860	192.168.29.1	192.168.29.2	ICMP	114	Echo (ping) reply id=0x0003, seq=4/1024, ttl=255 (request in 14)

- Frame 6: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface eth0, id 0
- Ethernet II, Src: 50:00:00:01:00:01 (50:00:00:01:00:01), Dst: 50:00:00:07:00:04 (50:00:00:07:00:04)
- Internet Protocol Version 4, Src: 192.168.29.2, Dst: 192.168.29.1
- Internet Control Message Protocol

Unicast routing - case study 1

Will the packet be sent out to GW or directly to the host?

- **Source:** 192.168.1.12/24
- **Destination:** 192.168.1.253/24

Unicast routing - case study 2

Will the packet be sent out to GW or directly to the host?

- **Source:** 192.168.1.12/24
- **Destination:** 192.168.2.253/24

Unicast routing - case study 3

Will the packet be sent out to GW or directly to the host?

- **Source:** 192.168.1.12/20
- **Destination:** 192.168.10.253/20

Unicast routing - case study 3

Will the packet be sent out to GW or directly to the host?

- **Source:** 192.168.1.12/20

11000000.10101000.00000001.00001100

- **Destination:** 192.168.10.253/20

11000000.10101000.00001010.00001100

Default gateway

- Host acting as a router for all the traffic without more specific route destined outside of the source's local network

```
root@ubuntu22:~# ip route  
default via 192.168.100.1 dev eth0 proto static
```

```
C:\Users\system32>route print
```

Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.100.1	192.168.100.201	35

Multiple Gateways

- Host can have more than one gateway depending on the destination

```
root@ubuntu22:~# ip route add 6.6.6.0/24 via 192.168.100.200 dev eth0
root@ubuntu22:~# ip r
default via 192.168.100.1 dev eth0 proto static
6.6.6.0/24 via 192.168.100.200 dev eth0
```

Multiple Gateways - metrics

- Host can have more than one gateway to the same destination

```
root@ubuntu22:~# ip route add 6.6.6.0/24 via 192.168.100.201 metric 20
root@ubuntu22:~# ip route add 6.6.6.0/24 via 192.168.100.202 metric 30
root@ubuntu22:~# ip route | grep 6.6.6.0
6.6.6.0/24 via 192.168.100.200 dev eth0
6.6.6.0/24 via 192.168.100.201 dev eth0 metric 20
6.6.6.0/24 via 192.168.100.202 dev eth0 metric 30
```

- Metric can be treated as **cost** to reach the destination

Multiple Gateways - metrics

- **Default metrics:** Every OS has its own logic what value to assign as a metric for a given prefix depending on the source of information (different routing protocols, static, directly attached...)
- With static routing, user can assign metric of their choice to the prefix. Paths with lower metrics are preferred.

Multiple Gateways - metrics

- Windows host connected simultaneously through cable and over wifi:

Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	192.168.100.1	192.168.100.201	35
0.0.0.0	0.0.0.0	192.168.100.1	192.168.100.247	25

- Guess which is which ;)

Static routing - overlapping networks

- Where will the packet toward **6.6.0.1** be sent?

```
root@ubuntu22:~# ip r
default via 192.168.100.1 dev eth0 proto static
6.6.0.0/25 via 192.168.100.202 dev eth0
6.6.0.0/24 via 192.168.100.201 dev eth0
6.6.0.0/23 via 192.168.100.203 dev eth0
```

Static routing - overlapping networks

- Where will the packet toward **6.6.0.1** be sent?

```
root@ubuntu22:~# ip r
default via 192.168.100.1 dev ens160 proto static
6.6.0.0/25 via 192.168.100.202 dev ens160
6.6.0.0/24 via 192.168.100.201 dev ens160
6.6.0.0/23 via 192.168.100.203 dev ens160
```

- The most specific route toward destination is always preferred. That's the route with the highest number of network bits matching the bits in the destination address

Static routing - overlapping networks

- Where will the packet toward **6.6.0.1** be sent?
 - **6.6.0.1:** 00000110.00000110.00000000.00000001
 - **6.6.0.0/23:** 00000110.00000110.00000000.00000000
 - **6.6.0.0/24:** 00000110.00000110.00000000.00000000
 - **6.6.0.0/25:** 00000110.00000110.00000000.00000000

Static routing - overlapping networks and different metrics

- Where will the packet toward **6.6.0.1** be sent?

```
root@ubuntu22:~# ip r
default via 192.168.100.1 dev eth0 proto static
6.6.0.0/25 via 192.168.150.1 dev eth1 metric 10
6.6.0.0/24 via 192.168.200.1 dev eth2 metric 20
6.6.0.0/23 via 192.168.250.1 dev eth3 metric 1
```

Static routing - overlapping networks and different metrics

- Where will the packet toward **6.6.0.1** be sent?

```
root@ubuntu22:~# ip r
default via 192.168.100.1 dev eth0 proto static
6.6.0.0/25 via 192.168.150.1 dev eth1 metric 10
6.6.0.0/24 via 192.168.200.1 dev eth2 metric 20
6.6.0.0/23 via 192.168.250.1 dev eth3 metric 1
```

- That would be again the /25 network as it is the longest match. Metrics come into play only if the target networks have 1:1 match!

Static routing - recursive routing

- Can the gateway IP belong to a host outside of any of the local networks?

Static routing - recursive routing

- Can the gateway IP belong to a host outside of any of the local networks?

IT CAN

Static routing - recursive routing

```
C:\Windows\system32>route ADD 6.6.0.0 MASK 255.255.255.0 192.168.100.1  
OK!
```

...

Network	Destination	Netmask	Gateway	Interface	Metric
6.6.0.0		255.255.255.0	192.168.100.1	192.168.100.247	26

```
C:\Windows\system32>route ADD 7.7.0.0 MASK 255.255.255.0 6.6.0.1  
OK!
```

...

Network	Destination	Netmask	Gateway	Interface	Metric
7.7.0.0		255.255.255.0	6.6.0.1	192.168.100.247	26

```
C:\Windows\system32>tracert -d 7.7.1.1  
Tracing route to 7.7.1.1 over a maximum of 30 hops  
1    1 ms    1 ms    1 ms  192.168.100.1  
2    8 ms    5 ms   10 ms  ...
```


Static routing - no gateway IP

- Can the static route point only through the interface without specifying an IP address?

Static routing - no gateway IP

- Can the static route point only through the interface without specifying an IP address?

IT CAN

Static routing - no gateway IP

```
root@ubuntu22:~# ip route add 8.8.0.0/24 dev eth0  
root@ubuntu22:~# ip r | grep 8.8.0.  
8.8.0.0/24 dev eth0 scope link
```

- Recap: if host doesn't know the MAC address of the host in the attached network it will send out ARP to resolve target IP to MAC address
- If a given host is reachable through the interface (without gateway IP) it will treat it as it belongs to the local network. Hence the ARP will be sent.
- ARP is broadcast, everyone in the local network will get this request.
- Device acting as router in the network would need to support **proxy-arp** functionality. It means it will respond to every ARP request with its MAC address